Astronomical Data Analysis Software and Systems XVIII
ASP Conference Series, Vol. 411, © *2009*
D. Bohlender, D. Durand and P. Dowler, eds.

A Comparison of Lossless Image Compression Methods and the Effects of Noise

W. D. Pence

NASA Goddard Space Flight Center, Greenbelt, MD, U.S.A.

R. Seaman

National Optical Astronomy Observatory, Tucson, AZ, U.S.A.

R. L. White

Space Telescope Science Institute, Baltimore, MD, U.S.A.

Abstract. This paper compares two image compression methods and shows how they are affected by the amount of noise in the image. We use a simple procedure to measure the equivalent number of noise bits in an image, which sets an upper limit on the compression ratio. We compare the widely used GZIP compression program to a newer compression method that uses the Rice algorithm within the FITS tiled-image compression convention. When applied to a large sample of astronomical images, the Rice compression method produces 1.4 times better compression, has 2–3 times faster compression speed, and has about the same uncompression speed as GZIP. In order to promote wider use of this new compression method, our `fpack` and `funpack` image compression software is available from the HEASARC and NOAO websites.

1. Introduction

As the size of astronomical data archives continues to increase exponentially, it is in the interests of both data providers and users to make the most effective use of the available image compression techniques. In this study we compare the commonly used technique of simply compressing FITS images using the GZIP utility program to the newer FITS tiled-image compression convention (Pence et al. 2000) using the Rice algorithm. Under this FITS convention, the image is first divided into a rectangular grid of "tiles" (usually on a row by row basis), and then each tile of pixels is compressed and stored in a variable length array column in a FITS binary table. A variety of different compression algorithms are currently available under this convention (and in principle any other algorithm could be supported), however we only report here on the results using the Rice algorithm, which is known to be both fast and produce high compression ratios.

Some of the advantages of this tiled image compression convention are

- the FITS header keywords remain uncompressed allowing fast access,
- the compressed FITS image is itself a valid FITS file,
- individual images in a multi-extension FITS file can be read without having to uncompress the entire file, and

- when reading a small section of the image, only the corresponding tiles need to be uncompressed.

In these comparison tests we used a complete sample of 1632 FITS 16-bit integer images that were taken during the night of 27–28 July 2006 at Cerro Tololo Inter-American Observatory using the Mosaic CCD camera. This data set contains a variety of different types of images that are typically taken during an observing session, including 0 s bias frame images, heavily exposed flat-field images of the interior of the dome, short exposures (10 s – 30 s) of bright calibration stars, and longer exposures (500 s – 600 s) containing randomly distributed images of stars and faint galaxies and patches of diffuse emission.

2. Noise Limits on Lossless Image Compression

The maximum possible image compression ratio that can be achieved by lossless compression algorithms is fundamentally limited by the well-known fact that a stream of randomly ordered bits is inherently incompressible. Thus, if the lowest N bits in each pixel of a hypothetical image are filled with noise, the maximum possible compression ratio is BITPIX/N, assuming that all the remaining non-noise bits can be compressed to zero. In general, no real algorithm can reach this limit, so the actual compression ratio will be given by BITPIX/$(N + K)$, where K is a measure (in units of bits per pixel) of how well, on average, the algorithm compresses the non-noise bits in each pixel of the image. The noise in real astronomical images is usually not neatly confined to the lowest N bits of each pixel; in the CCD images used in this study, the pixel values close to the the background level can be closely approximated as having a Gaussian noise distribution. It can be shown that the *equivalent* number of noise bits per pixel in an image with Gaussian-distributed noise is given by $N_{bits} = \log_2(\sigma\sqrt{12}) = \log_2(\sigma) + 1.8$, where σ is the Gaussian standard deviation of the pixel values.

3. Comparison of Compression Methods

We compressed and uncompressed each of the Mosaic camera CCD images in our sample using two different methods: (1) the Rice algorithm within the FITS tiled image compression convention, and (2) the familiar GZIP utility program. In each case we recorded the compression ratio and the elapsed compression and uncompression CPU times. We also calculated the Gaussian σ of the noise in the local background regions of each image using a robust algorithm originally developed to measure the signal to noise in spectroscopic data (Stoehr et al. 2007). From this we then calculated the equivalent number of noise bits per pixel in the image.

The primary result of this study is given in Figure 1, which shows the compression ratio achieved when using the Rice and GZIP methods plotted as a function of the measured equivalent number of noise bits in each image. The main conclusions can be summarized as follows:

- The Rice algorithm (the upper set of points) produces about 1.4 times better compression than GZIP (lower points).
- The points are segregated into three distinct groups corresponding to the different types of images: the bias frame images, the short exposures of

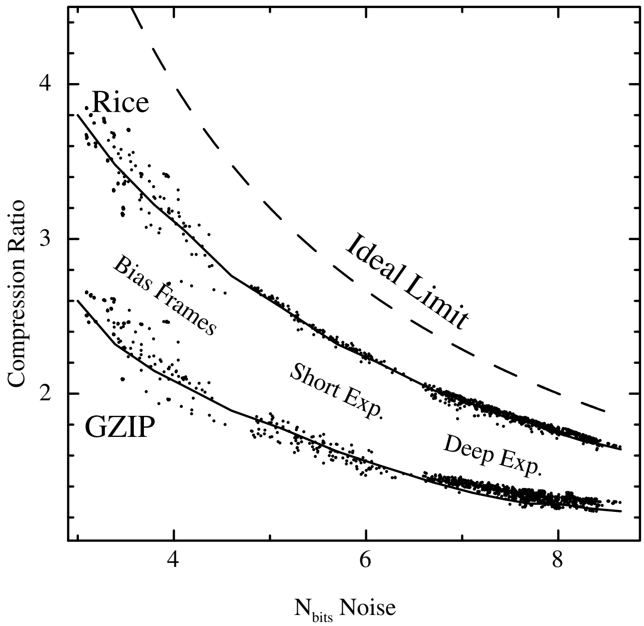

Figure 1.: Image compression ratios produced by Rice and GZIP as a function of the equivalent number of noise bits in the image.

bright calibration stars, and the deeper exposures of the sky and flat fields. This grouping is simply due to the fact that the different types of images have different mean pixel count levels, and the noise from Poisson statistics is proportional to the square-root of the mean pixel value.

- The real astronomical images closely follow the same tight correlation between noise and compression ratio that is derived from a set of synthetic flat field images that were constructed with differing amounts of pure Gaussian-distributed noise (shown by the curve going through each set of points). This demonstrates that the objects in these astronomical images (e.g., the images of stars and faint galaxies) have very little effect on the overall compression ratio of the image. This is not too surprising because most of the pixels in these images have values close to the local background level, and the astronomically interesting objects typically only cover a few percent of the total image area. Thus, even if the algorithms do not compress the regions close to these bright objects very effectively (due to larger variations in the pixel values), the overall compression ratio of the image will still mainly just depend on the compressibility of the background regions.
- The upper dashed curve shows the theoretical maximum compression ratio, given by BITPIX/N_{bits}, that would be produced by an ideal lossless algorithm that infinitely compresses all the non-noise bits in the image and has $K = 0$. The Rice algorithm is not far from this, with a K value of about 1.2 bits per pixel, and thus achieves about 75% to 90% of the

ideal compression ratio, depending on the noise level. This implies that there is relatively little more to be gained from using algorithms that have even smaller K values. By comparison, GZIP has a much larger effective K value, ranging from about 2.5 to 4.5 bits per pixel, depending on the amount of noise in the image.

As well as producing higher compression ratios, our CPU timing tests showed that the Rice algorithm is 2–3 times faster than GZIP when compressing these images (the difference is greatest when compressing the noisiest images). There was less of a difference when uncompressing the images however. The uncompression times are nearly identical for the noisier, deeply exposed images, while GZIP is about 20% faster then uncompressing the lowest noise images. As a bench mark reference, a Linux PC with a 2.4 GHz AMD Opteron 250 dual core processor (using only one of the processors) can Rice compress a 50 MB 16-bit integer FITS image in one second. Uncompressing this image also takes about one second.

4. Summary

These tests show that the Rice algorithm produces much higher compression ratios and is also faster in most cases than GZIP. The FITS tiled-image file format also has many advantages over simply compressing the whole file with GZIP and is transparently supported by major software packages such as the CFITSIO library and the ds9 image display program. To help promote the use of this new image compression method, our open-source image compression utilities, called `fpack` and `funpack`, (Seaman et al. 2007) are publicly available from the HEASARC web site[1] for general use. These programs run on all major computer platforms and are invoked on the command line just like GZIP and GUNZIP to compress or uncompress any FITS image file. Various options can be specified on the command line to control the programs. More information about using `fpack` and `funpack` is available in the companion user's guide.

References

Pence, W. D., White, R. L., Greenfield, P., & Tody, D. 2000, in ASP Conf. Ser. 216, ADASS IX, ed. N. Manset, C. Veillet, & D. Crabtree (San Francisco: ASP), 551

Seaman, R., Pence, W. D., White, R. L., Dickison, M., Valdes, F., & Zarate, N. 2007, in ASP Conf. Ser. 376, ADASS XVI, ed. R. A. Shaw, F. Hill, & D. J. Bell (San Francisco: ASP), 483

Stoehr, F. et al. 2007, ST-ECF Newsletter, 42, 4

[1] http://heasarc.gsfc.nasa.gov/fitsio/fpack